DOCUMENT RESUME

ED 262 758 IR 011 836

TITLE	ICCE Policy Statement on Network and Multiple Machine Software.
INSTITUTION	International Council for Computers in Education, Eugene, Oreg.
PUB DATE	[85]
NOTE	16p.
AVAILABLE FROM	International Council for Computers in Education, University of Oregon, 1787 Agate Street, Eugene, OR 97403.
PUB TYPE	Legal/Legislative/Regulatory Materials (090) -- Guides - Non-Classroom Use (055)
EDRS PRICE	MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS	*Computer Software; *Copyrights; Guidelines; Legal Responsibility; *Microcomputers; School Districts; School Policy
IDENTIFIERS	*Computer Uses in Education; *Software Copying

ABSTRACT

Designed to provide educators with guidance for the lawful reproduction of computer software, this document contains suggested guidelines, sample forms, and several short articles concerning software copyright and license agreements. The initial policy statement calls for educators to provide software developers (or their agents) with a district-level approved written policy statement which includes the responsibilities of both the educator and the school, and lists the responsibilities of the hardware vendor and the software developer/vendor in assisting educators to observe copyright laws and publishers' license agreements. Four attachments contain a suggested school district policy on software copyright, a software policy for a community college with a large microcomputer lab, a suggested format for software licensing, and some technical notes on software encryption for software/hardware vendors. Four articles on computer software copyright conclude the document: "Software Copyright Interpretation" (LeRoy Finkel); "Computers, Software and the Law" (David G. Novick); and "Right and Wrong in Educational Computing" (David G. Novick). (JB)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
This document has been reproduced as received from the person or organization originating it.
Minor changes have been made to improve reproduction quality.
Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

Publishers of **The Computing Teacher** journal.

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY
David Morsund
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

ICCE Policy Statement on Network and Multiple Machine Software

Just as there has been shared responsibility in the development of this policy, so should there be shared responsibility for resolution of the problems inherent in providing and securing good educational software. Educators have a valid need for quality software and reasonable prices. Hardware developers and/or vendors also must share in the effort to enable educators to make maximum cost-effective use of that equipment. Software authors, developers and vendors are entitled to a fair return on their investment.

ED262758

Educators' Responsibilities

Educators need to face the legal and ethical issues involved in copyright laws and publisher license agreements and must accept the responsibility for enforcing adherence to these laws and agreements. Budget constraints do not excuse illegal use of software.

Educators should be prepared to provide software developers or their agents with a district-level approved written policy statement including as a minimum:

1. A clear requirement that copyright laws and publisher license agreements be observed;
2. A statement making teachers who use school equipment responsible for taking all reasonable precautions to prevent copying or the use of unauthorized copies on school equipment;
3. An explanation of the steps taken to prevent unauthorized copying or the use of unauthorized copies on school equipment;
4. A designation of who is authorized to sign software license agreements for the school (or district);
5. A designation at the school site level of who is responsible for enforcing the terms of the district policy and terms of licensing agreements;
6. A statement indicating teacher responsibility for educating students about the legal, ethical and practical problems caused by illegal use of software.

Hardware Vendors' Responsibilities

Hardware vendors should assist educators in making maximum cost effective use of the hardware and help in enforcing software copyright laws and license agreements. They should as a minimum:

1. Make efforts to see that illegal copies of programs are not being distributed by their employees and agents;
2. Work cooperatively with interested software developers to provide an encryption process which avoids inflexibility but discourages theft.

Software Developers'/Vendors' Responsibilities

Software developers and their agents can share responsibility for helping educators observe copyright laws and publishers' license agreements by developing sales and pricing policies. Software developers and vendors should as a minimum:

1. Provide for all software a back-up copy to be used for archival purposes, to be included with every purchase;
2. Provide for on-approval purchases to allow schools to preview the software to ensure that it meets the needs and expectations of the educational institution. Additionally, software developers are encouraged to provide regional or area centers with software for demonstration purposes. The ICCE encourages educators to develop regional centers for this purpose;

IR011836

University of Oregon 1787 Agate St. Eugene, OR 97403 USA 503/686-4414

3. Work in cooperation with hardware vendors to provide an encryption process which avoids inflexibility but discourages theft;
4. Provide for, and note in advertisements, multiple-copy pricing for school sites with several machines and recognize that multiple copies do not necessarily call for multiple documentation;
5. Provide for, and note in advertisements, network-compatible versions of software with pricing structures that recognize the extra costs of development to secure compatibility and recognize the buyer's need for only a single copy of the software.

The Board of Directors of The International Council for Computers in Education approved this policy statement, with attachments, June 5, 1983.

The committee that drafted this policy included:

Jenny Better, Director of Curriculum, Cupertino Union Elementary District
LeRoy Finkel, San Mateo County Office of Education
Pennie Gallant, Apple Computer, Inc.
John Hazelwood/Jeffrey Armstrong, Corvus Systems, Inc.
Marion B. Kenworthy, Saratoga High School
Richard R. Monnard, Addison-Wesley Publishing Co.
Henry Vigil/Cliff Godwin, Cybertronics International
William Wagner, Santa Clara County Office of Education

University of Oregon 1787 Agate St. Eugene, OR 97403 USA 503/686-4414

Publishers of **The Computing Teacher** journal.

ATTACHMENT 1
Suggested District Policy on Software Copyright

It is the intent of _____ to adhere to the provisions of copyright laws in the area of microcomputer programs. Though there continues to be controversy regarding interpretation of those copyright laws, the following procedures represent a sincere effort to operate legally. We recognize that computer software piracy is a major problem for the industry and that violations of computer copyright laws contribute to higher costs and greater efforts to prevent copies and/or lessen incentives for the development of good educational software. All of these results are detrimental to the development of effective educational uses of microcomputers. Therefore, in an effort to discourage violation of copyright laws and to prevent such illegal activities:

1. The ethical and practical problems caused by software piracy will be taught in all schools in the District.
2. District employees will be expected to adhere to the provisions of Public Law 96-517, Section 7(b) which amends Section 117 of Title 17 of the United States Code to allow for the making of a back-up copy of computer programs. This states that "... it is not an infringement for the owner of a copy of a computer program to make or authorize the making of another copy or adaptation of that computer program provided:
   a. that such a new copy or adaptation is created as an essential step in the utilization of the computer program in conjunction with a machine and that it is used in no other manner, or
   b. that such a new copy and adaptation is for archival purposes only and that all archival copies are destroyed in the event that continued possession of the computer program should cease to be rightful."
3. When software is to be used on a disk sharing system, efforts will be made to secure this software from copying.
4. Illegal copies of copyrighted programs may not be made or used on school equipment.
5. The legal or insurance protection of the District will not be extended to employees who violate copyright laws.
6. _____ of this school district is designated as the only individual who may sign license agreements for software for schools in the district. (Each school using the software also should have a signature on a copy of the software agreement for local control.)
7. The principal of each school site is responsible for establishing practices which will enforce this policy at the school level.

Publishers of **The Computing Teacher** journal.

ATTACHMENT 2

Sample Software Policy of a Community College with a Large Microcomputer Lab

It is the policy of this college that no person shall use or cause to be used in the college's microcomputer laboratories any software which does not fall into one of the following categories:

1. It is in the public domain.
2. It is covered by a licensing agreement with the software author, authors, vendor or developer, whichever is applicable.
3. It has been donated to the college and a written record of a bona fide contribution exists.
4. It has been purchased by the college and a record of a bona fide purchase exists.
5. It has been purchased by the user and a record of a bona fide purchase exists and can be produced by the user upon demand.
6. It is being reviewed or demonstrated by the users in order to reach a decision about possible future purchase or request for contribution or licensing.
7. It has been written or developed by _____ (college employee) for the specific purpose of being used in the _____ (college) microcomputer laboratory.

It is also the policy of the college that there be no copying of copyrighted or proprietary programs on computers belonging to the college.

Source: De Anza College, Cupertino, California.

Publishers of **The Computing Teacher** journal.

ATTACHMENT 3
Suggested Format of Software Licenses

1. Designated on a per site, district-wide or other geographic basis.
2. Requires the signature of a responsible school employee.
3. Includes provisions for a single copy purchase (with archival back-up copy) at full price.
4. Multiple-machine pricing:
   Includes provisions for a quantity discount for subsequent purchases of the same software provided:
   a. the purchase discount applies to a single purchase order.
   b. the purchase discount is noncumulative.
   c. the software is for the same computer type.
   i.e.: Radio Shack presently offers a 50% discount for purchases of 10 or more sets of the same software; Gregg/McGraw-Hill offers a discount schedule with incremental increases—buy 2, pay 10% less; 3—20% less; 4—30% less; 5 or more, 40% less.
5. Network Pricing:
   May be offered as per school site or with quantity discount for school districts with multiple sites.
   Provide for a flat license fee for network-compatible versions of the software.
   - flat fee provision is preferred over any variable rate based on number of computers or number of student users.
   - network-compatibility, not just an unlocked version of the software, is required to eliminate the need for local reprogramming of copyrighted and licensed software.

   Include provision for purchase of multiple copies of documentation and accompanying materials.
   i.e.: A flat fee of two times the single copy retail price is offered to network users of Random House software.

University of Oregon 1787 Agate St. Eugene, OR 97403 USA 503/686-4414

Publishers of **The Computing Teacher** journal.

ATTACHMENT 4
Some Technical Notes on Software Encryption for Software/Hardware Vendors

1. Single Machine Encryption

Explanation:

The purchased disk is not copiable by ordinary means. The software cannot be transferred to a network system or used on several computers at once. This scheme is the most common, especially for inexpensive software.

Technical notes:

The protected disk is usually formatted in a non-standard way which will defeat standard disk copy programs such as COPYA on the Apple or TRSDOS BACKUP on the TRS-80. Alternatively, the publisher may write special information on the disk in places which the standard disk copy programs do not check. The copy program proceeds to completion, but the special information is not transferred to the duplicate disk. When the duplicate is used, the software checks for the special information, fails to find it, and stops.

Implications:

Schools will need to purchase many copies of the same program and should expect significant volume discounts. The customer is entitled to an archival backup and should expect the publisher to include a backup disk with every purchase.

Manufacturers of network systems should recognize that single machine encryption (which is incompatible with their products) will remain the software industry standard unless they actively support software protection on their systems.

2. Single Site Encryption

Explanation:

A single product can serve all the machines at a site. This scheme applies to VisiCalc™ and Logo.

Technical Notes:

Software which loads initially into memory and subsequently interacts only with data disks is de facto "single site encrypted," even though the program disk may be uncopiable. A single program disk can be used to initialize all the computers in a room, after which each user operates with his or her own data disks. VisiCalc™ and Logo operate in this way.

A functionally equivalent alternative is referred to as "master and slave" or "lock and key" encryption. This scheme is common where a program is too large to fit in memory all at once. Frequent disk access is needed as different parts of the software are brought into play.

In the "lock and key" scheme, the program modules which are routinely needed can be freely copied. A "slave" disk containing these modules is duplicated for each computer (or even for each student). The slave will not operate, however, unless the computer has been cold started with the (uncopiable) master disk.

Implications:

Since the "master" disk is uncopiable, the publisher still bears the burden of providing an archival backup. The protection on the "master" disk normally makes the software incompatible with network systems, so the above comments again apply.

Single site encryption reduces the dependence on volume discounts to facilitate multiple machine use. However, volume discounts should still be made available at the district level to encourage district level adoption of software.

3. **Hard Disk/Network Compatible Versions of Software**

Explanation:

Floppy disks containing network compatible software must be copiable since the software is copied as it is transferred onto the network. The problem of protecting network compatible software is how to allow this legitimate copying while preventing illegal copying.

One solution is to abandon software protection altogether and to rely on license agreements to prevent illegal use of the program(s). The problem with this solution is that freely copiable software may be freely copied.

Other solutions rely on publishing special versions of the software for the various network systems available. These versions do not run on stand-alone computers.

A publisher can also take steps to discourage people from installing the network software at sites other than the intended site.

Technical Notes:

A publisher can prevent network software from running on a stand-alone computer by using a device check. The software senses whether it is running on a network system and stops if it is not. The device check is specific to the network system involved. Software with a device check could be installed at many network sites, not just the one for which it was licensed.

To discourage use at non-licensed sites, the publisher can embed the name of the licensee in the software. This requires that the publisher customize each network-compatible version sold. Although such customization discourages porting the software to another network site, it does not physically prevent it.

To prevent porting of the software to another network, the publisher might implement what is essentially single machine encryption on the network level. This protection scheme would work by checking the serial number or other unique identifier in the network hardware. If the software encountered a change in identifier, it would fail to operate. This has the disadvantages that a licensee would have to be a single network installation and that normal activities such as replacing or upgrading one's network system would disable the software.

Implications:

Use of a device check or serial number check requires a publisher to maintain a separate inventory item for each device to be supported. The time required for a publisher to embed the customer's name in each product sold for use on networks can become prohibitive.

These protection schemes may prove economically unfeasible for inexpensive software.

These protection schemes require close working relationships and sharing of information between publishers and network system manufacturers.

Software Copyright Interpretation

by
LeRoy Finkel

I have been asked to reconvene the ICCE Software Policy Committee so that we may examine the current state of the art (things in law tend to change over time) and review our current policies. As we prepare to meet, it seems appropriate to share with *The Computing Teacher* readers the best information we have regarding current interpretation of copyright laws.

There are no definitive answers to most of the questions we have, since the copyright law is vague in places and there have been no court cases to set precedent. Nevertheless, copyright attorneys, court watchers and lawmakers all seem to agree on how a court would interpret the current law if and when a case came before it. Not wanting to get sued and wanting to encourage software development by vendors, I prefer to take a conservative approach that looks to how the law will *likely* be interpreted, rather than waiting for the definitive decision by a court. In other words, I don't want to be the test case!! Do *you*? For those who doubt that publishers will sue a school district or teacher, be reminded that the American Association of Publishers did sue New York University, that a BOCES in New York was also sued (both public agencies lost their cases), and that while publishers may not sue, their professional associations seem willing and able to do so.

The issues:

1. **Back-up copies.** You are allowed back-up copies (number uncertain) that are to be used for archival purposes in the event your original copy fails. Such copies are *not* to be used on a second machine at the same time as the original. Since a backup is allowed by law, and if your vendor does not provide one or allow a process by which you can acquire one, then you may make one. But its *use* is restricted as stated above. Vendors who offer "multiple" back-up copies are using the term "back-up" incorrectly and have been asked to use the term "multi-copy discounts," which more accurately reflects what they are offering you.
2. **Multiple-loading or booting from one disk into multiple machines at the same time.** "In the absence of a license that explicitly permits you to do so," you would likely be in violation of the copyright laws if you loaded multiple computers with the contents from one disk for use at the same time. The legal concept has to do with the "proliferation of simultaneous users." The law is designed to protect the copyright holder from loss of sales. If *Bank Street Writer* is sold for use on one machine (and it is), and you load it into 15 machines, one after the other, so that all 15 are in use at the same time, you are inhibiting sales. Thus, you are in violation of the law. The fact that you *can* physically load the contents into multiple machines is irrelevant. The law does imply that sequential use on different machines is okay (first on one machine, turn it off, then on another machine). The key element here is proliferation of "simultaneous" users. That one concept has helped me out a lot. Two companies have recently announced simultaneous-use or multiple-loading software. They have been asked to emphasize that this is a special license for a particular piece of software. One solution to the multiple-loading "problem" is multi-copy pricing and licensing, an option more companies seem to be taking.
3. **Networks.** "In the absence of a network license" you would likely be in violation of copyright laws if you downloaded a program to multiple stations at the same time from your network, be it a hard disk or floppy disk network. The "proliferation of simultaneous users" concept described above would again apply. *Whether it is physically possible to load the stations from the network is not germaine to this discussion. The absence of a license permitting simultaneous use is the copyright issue.*

It is not enough for districts to merely pass copyright policies—we must pay heed to them. It is the responsibility of each of us to be a role model to fellow teachers and students alike and allow only legal uses of software on our campuses.

If you have questions, comments or information for the committee, please write me. Since the law is somewhat different in each country, I would like to hear from people willing to serve on subcommittees for specific countries.

[LeRoy Finkel, San Mateo County Office of Education, 333 Main Street, Redwood City, CA 94063.]

Computers, Software and the Law

by
DAVID G. NOVICK

SIGOC is pleased to introduce a continuing column entitled "Computers, Software and the Law," written by attorney David G. Novick. Mr. Novick welcomes your questions regarding legal problems with software under the laws of the United States. Readers' questions will be answered, space permitting. If you have a question about copyright or related topics, please send it to "Computers, Software and the Law," c/o ICCE SIGOC, University of Oregon, 1787 Agate St., Eugene, Oregon 97403, USA.

The answers given in this column are a discussion of general principles of American law, and are intended to provide background information for people who work with computers and software. The author neither purports nor undertakes to represent persons submitting questions. If you have a specific legal problem, please consult with a qualified lawyer in your area.

Question: If I buy a copy-protected program, is it OK for me to copy it with a copying program in case the original is damaged?

Answer: Making a copy of a computer program for archival purposes does not infringe the author's copyright for the program. Permission to make such copies is given explicitly by Sec. 117 of the Copyright Act (17 U.S.C. Sec. 117). However, there are significant limitations and conditions on this permission. In particular, the statute provides that it is not an infringement for you to make or to authorize the making of another copy or adaptation of the program only if 1) you own the copy from which the new copy is made; 2) the new copy or adaptation is for archival purposes only; and 3) all archival copies are destroyed in the event that you no longer rightfully own the first copy from which the archival copy was made.

Be sure to note, too, that copyright law may not be the only legal barrier to making back-up copies. Vendors of computer software do not usually sell their programs outright. Typically, software publishers sell only a license to use the program. That is, the purchaser, who may be called the "licensee" in an agreement, buys only a right to use the program rather than any ownership interest in the program itself. Some licenses even provide that the physical disk on which the user's copy is distributed remains the property of the seller.

Many licenses restrict the right of the buyer-user, among other things, to make copies. If you have a license agreement or other contract from a software supplier, read the terms of the contract before making any copies. (In fact, it is a good idea to read the contract before buying the program.) Therefore, although making an archival copy of a program may not violate the Copyright Act, the making of the copy may still be prohibited by your agreement with the vendor of the program.

Question: I'm a teacher and I've been working on a program that I'd now like to market. My principal says the school district has a right to the royalties from the program. Is this true?

Answer: Maybe. The answer depends on a number of factors, including the nature of the program, your duties as an employee of the school or school district, the use of the program, and how, when and where you wrote the program. The general rule is that the copyright in a program belongs to the "author" of the program. Unfortunately, it is sometimes difficult to determine who is the "author" for the purposes of the Copyright Act. Just because you are employed by someone else does not mean that the other person owns the rights to the program you wrote. However, your employer is automatically considered to be the author of the program if the program was written "for hire." A program is written for hire if 1) you are employed and 2) the program was written within the scope of your employment. In simpler terms, the program belongs to your employer if you wrote the program as part of your job. (A work is also deemed made "for hire" under other circumstances, particularly where the program's creator is not an employee and he or she writes the program in response to a special order or commission. These provisions are found at 17 U.S.C. Sec. 101.)

Therefore, in determining whether the program was written in the course of your employment, you and your employer should assess the following factors:

1) Did your employer have the right to tell you to work (and how to work) on the program?
2) Were you being paid by the school for the hours you actually spend developing the program?
3) What use, if any, did you make of the school's facilities in developing the program?

Note, though, that writings by government employees relating to their employment, written mostly during off-duty hours, are not government property. In an older case, it was held that a teacher's

printed materials are well known and generally respected. The law concerning networking or booting a disk in eight different machines is not as clear. If the original copy of a disk is ruined, can we make a backup copy of the backup copy?**

All of these issues relate to how we view a software policy. Coordinators were unsure on how to act or react to these problems.

From my survey, I was not able to identify a sound software protection policy. Most schools and school districts felt that a strengthening of their policy was necessary, but did not know exactly how to proceed.

Software protection is a major concern of all educators, and even though I have not presented the guide I had intended, perhaps this article will generate some response. If you have questions, please ask them. If you have a workable plan or ideas for such a plan, share them! This is your newsletter—other coordinators want to hear what you have to say.

[Bernie Nelsen is a graduate student in computer education at the University of Oregon. He has taught secondary math for fourteen years in Iran and Thailand as well as in the United States.]

*The IOCE Policy Statement is in the September 1983 issue of The Computing Teacher and is also available on request. Please send a self-addressed stamped envelope to IOCE Software Policy Statement, University of Oregon, 1787 Agate St., Eugene, OR 97403. Permission is granted by IOCE to make copies of the policy statement.

**For answers to some questions on copyright and software, see David Novick's article in this issue, page 5.

Software Protection Policies

[The following policy statement was provided by Julie Willey, Program Director for Community Youth Creative Learning Experience (C.Y.C.L.E.) in response to Bernard Nelsen's article, "Software Protection Policy—What Is It?" in the introductory issue of the SIG Bulletin.*]*

POLICY STATEMENT on MICRO ETHICS

1. Ethical, as well as practical problems caused by software piracy will be taught.
2. No illegal copies of copyrighted programs will be made or used on C.Y.C.L.E. equipment.
3. Legal copies of copyrighted programs will be protected to prevent unauthorized additional copies.
4. C.Y.C.L.E. stands behind the authors, companies and professional community in upholding the copyright laws for software.
5. C.Y.C.L.E. will accept no copies of software that are *knowingly* illegal, regardless if they are donated or obtained in some other way.
6. The legal or insurance protection of C.Y.C.L.E. will not be extended to staff, students and others who violate copyright laws.
7. It is the intent of C.Y.C.L.E. to adhere to the provisions of copyright laws in the area of microcomputer software. This means compliance of Public Law 96-517, Section 7(b) which amends Section 117 of Title 17 of the United States Code to allow for the making of a back-up copy of computer programs. This states that "... it is *not* an infringement for the owner of a copy of a computer program to make or authorize the making of another copy or adaptation of that computer program provided:
   a. that such a new copy or adaptation is created as an essential step in the utilization of the computer program in conjunction with a machine and it is no other manner, or
   b. that such a new copy and adaptation is for archival purposes only and that all archival copies are destroyed in the event that continued possession of the computer program should cease to be rightful."

Computers, Software and the Law

by
DAVID G. NOVICK

[The answers given in this column are a discussion of general principles of American law, and are intended to provide background information for people who work with computers and software. The author neither purports nor undertakes to represent persons submitting questions. If you have a specific legal problem, please consult with a qualified lawyer in your area.]

Question: What is the current status of laws regarding the copying (pirating) of software that has a copyright? I have heard that it is legal to make back-up copies. I have also heard that the whole question of copyright for software is legally "up in the air."

Answer: The law of copyright for software is now much clearer than it was before 1976. For the first time, programs, even in machine-readable form, are copyrightable. (Actually, the law now considers computer programs "non-dramatic literary works," so hackers are now in the company, legally speaking, of Twain, Proust, and Kerouac.) It is permissible to make back-up copies, as discussed above. Therefore, copying and pirating are not synonymous. Pirating is the making, and sometimes distributing or selling, of illegally-made copies. Pirating of programs is infringement of copyright, and makes the pirate both civilly and criminally liable under copyright laws. Other sanctions for pirating, including prosecution for theft and civil actions for damages under state law, are also possible.

Question: A team of teachers in our school district is working on a set of programs which may have commercial value. The project will probably take

over a year to finish. We have a number of questions relating to the project. First, can we copyright the programs before they are completed? Should we copyright both the documentation for the project and the programs themselves? What kind of protection will a copyright give us? Finally, would you outline the general steps involved in obtaining a copyright?

Answer: Under the copyright laws of the United States, a computer program (or any other work, for that matter) can be copyrighted as soon as it is in a "fixed" form. This means that a program can be copyrighted when it is written or typed on paper, stored on disk or tape, or otherwise stored in a manner in which the program may be read by a person or a machine. The authors' expectations that the program will later be revised or expanded does not affect the copyrightability of the program. Rather, the authors can subsequently copyright the changes and additions as they are made. Therefore, you will probably want to obtain copyright protection for the program as soon as you create it. Note that the length of protection for programs which are jointly created is the life of the longest-surviving author plus fifty years. However, where the program is written by employees for the benefit of their school district as part of their jobs, the program is considered a work written "for hire," so the term of protection is 100 years from the date of creation of the program, or 75 years from the date of publication of the program, whichever is shorter.

Both the program and the documentation should be copyrighted! Failure to obtain adequate copyright protection for either the program or the documentation may result in losing all or some rights in the work. This means that you should put appropriate copyright notices in both the documentation and the program itself, and should comply with the registration and deposit requirements for both.

A copyright is just that: the exclusive right to make copies of a computer program or other work. The copyright act provides that the owner of a copyright has the exclusive rights to do and to authorize any of the following things:

1) To reproduce the program in copies;
2) To prepare the derivative works based upon the program;
3) To distribute copies of the program to the public by sale or other transfer of ownership, or by rental, lease or lending.

There are also some other rights which mainly involve non-computer works, such as plays and movies.

You should be aware that some aspects of protection depend on timely compliance with the registration and deposit requirements. The most significant aspect, especially for smaller businesses, is the right to collect your attorney fees from the infringer if you win an infringement claim. Therefore, register your program within three months of publication of the program. There are special rules dealing with how many and what kinds of copies of the program have to be sent in with the registration. Generally, you should deposit one copy of the first and last 25 pages of your program listing, and two good copies of the documentation. You can and should consult a recent book on computer copyrights for details; if the project is complicated or especially important, consult an attorney.

You still have these rights for the length of the applicable copyright term. As a consequence, you will have a right to collect damages for infringement from other persons who make copies of your program, whether they sell them or not. There are, of course, some exceptions to this rule; these are mainly aspects of "fair use," and are rather fluid and technical. With respect to computer programs in particular, the owner of a copy of the program may make archival copies, and can make "copies" of the program in computer memory which may be necessary in order to run the program. If you need more information about this aspect of copyright protection, I suggest that you consult an attorney with knowledge in the field.

The mechanics of obtaining a United States copyright are relatively simple. Make sure that the copyright notice "Copyright 1984 Your Name Here" is at the beginning, title page (if any), and end of the program listing, that the program displays the copyright notice either when it starts or throughout the program, and that the packaging of the disk or cassette also displays the copyright notice. Actually, any of the methods alone is sufficient under the Copyright Act, but I suggest that you use all of them in order to make absolutely sure that the notice is valid and that users understand clearly that the program is copyrighted. To register the program, obtain Form TX from the United States Copyright Office, Library of Congress, Washington, D.C. 20559; fill out the form as directed, and return it along with a check or money order for $10 and the requisite number of copies.

Question: A group of my students has studied the ideas and coding of a gradebook program. They like the general format and style of the program, but believe that they could make it significantly easier to enter data and also speed up its routines significantly. If my students rewrite the program and improve its performance substantially while keeping the general ideas, can they copyright it?

Answer: I see two potential problems here. The first is whether the author of the original gradebook program has rights which would limit the protection available to your students. The second is whether the student's program has any aspects which are copyrightable in themselves.

In looking at the rights of the original author, you might start by checking if the gradebook program

was copyrighted. If not, the program is likely to be in the public domain, and so use of the program by your students would most probably not be an infringement. If the program were copyrighted, however, your students can still use the ideas embodied in the program, if not parts of the actual program. Perhaps the most significant limitation on copyright protection is that copyright protects only the expression of an idea and not the idea itself. (In contrast, patents protect the inventor's method itself, and not just how the method is written down.) The Copyright Act provides that, "In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work." This means that the students can use the system which the program implements if they write their own routines to do them.

The second aspect is a sort of mirror image of the first. Note that the students will only have rights in the code they write, and will not have rights to any method of keeping gradebooks. Moreover, if the original program turns out to have been in the public domain, the students' copyright will cover only the changes they made and will not protect parts of the old program incorporated into the new program.

[Attorney David G. Novick welcomes your questions regarding legal problems with software under the laws of the United States. Readers' questions will be answered, space permitting. If you have a question about copyright or related topics, please send it to "Computers, Software and the Law," c/o ICCE SIGCC, University of Oregon, 1787 Agate St., Eugene, OR 97403, USA.]

[Attorney David G. Novick earned his J.D. degree from Harvard Law School in 1977. He is a member of the bar of the District of Columbia, the Oregon State Bar, the Federal Communications Bar Association and the American Bar Association. In addition to his private practice, Mr. Novick has taught at the University of Oregon School of Law since 1981, where he is Adjunct Assistant Professor of Law. Mr. Novick is a founder and the managing partner of McKenzie InterMedia Group, a consulting firm emphasizing the relationship between technology and marketing. Mr. Novick is also in the Ph.D. program in Computer and Information Science at the University of Oregon.]

Right and Wrong in Educational Computing

by
DAVID G. NOVICK

[A preliminary version of this article appears in the proceedings of the "Computer Extension of the Human Mind" conference held in Eugene, Oregon in August 1984.]

What is wrong with copying other persons' programs if they won't be hurt? Teachers who use computers routinely face this and similar questions, but lack a systematic way to find answers. Some writers have proposed a professional code of ethics for computer science. However, for the reasons discussed below, codification of ethics may not be the best approach. Perhaps something like an informal etiquette would be appropriate. Given the nature of the field, the origins of our ideas about ethics, and the novelty of the questions we face, any approach to ethics in computer science should be developed on an evolving, case-by-case basis. Indeed, we should encourage students to participate in the development of such a system; in this way students may regard adherence to notions of right and wrong in computer science as a shared responsibility rather than an arbitrary and authoritarian burden.

For purposes of analysis, I propose to distinguish between *morality* as a measure of absolute goodness in some natural sense and *ethics* as a system or collection of precepts based on morality. In other words, *morality* is our knowledge of what is good, and *ethics* are systematic ways of resolving conflicts between competing goods. Given even this general framework, it would seem to be hard to construct a morality of computer science, let alone a system of ethics based on such a morality. There are, however, certain things upon which we can probably agree: it is wrong to injure someone; curiosity is good; putting knowledge to work is good; and so forth. One might object that "curiosity is good" (or any of the other tenets above) is not always true. This objection, I believe, is ill-founded because it confuses morals and ethics. That is, situations where one might like to inhibit curiosity arise not because the curiosity is itself bad but because some other moral value, perhaps privacy, seems more important under the circumstances. Given a set of agreed upon moral values (or, to put it another way, goals), we are faced with the problem of resolving situations where moral values conflict. This is what I have been referring to as *ethics*.

Using this definition, there are obviously a large number of ethical decisions to be made in computer science and computer education, but it is not obvious that any of these decisions are difficult ones. In medicine, by way of contrast, there are clearly difficult ethical decisions; for example, our moral sense that life should be preserved conflicts with our sense that persons should be spared misery when facing the plight of a terminally ill person in much pain whose life is being sustained through "heroic" efforts. Similarly difficult questions exist in other professions such as law. In the sciences and in scientific education, there has been little clamor for guidance on questions of similar difficulty. A survey of formal ethics in the sciences conducted in 1981 found that the few ethical codes which did exist were typically promulgated by engineering societies, that the codes had a fair amount of formality, and that there was neither widespread dissemination or acceptance in the scientific community. My own review of proposed codes of ethics in computer science suggests that such codes are chiefly concerned with upholding the dignified image of the profession which computer programmers aspire to create.

There are, nevertheless, some interesting issues resulting from moral conflict in the use of computers. Conflicts between the value of utility versus the value of an orderly society arise in the case of theft of software, copyright violations, and invasion of privacy. The value of curiosity is clearly at odds with the value of privacy in many situations, including the creation and use of large data bases. In some of these cases there are laws which provide relatively clear guides for behavior independent of ethical considerations. In other situations, the law is obscure or non-existent.

I propose, then, to look at the interaction of ethics and law in three specific situations which I believe are commonly encountered by computer educators. In one instance, there are relatively clear markers for acceptable conduct. In the other two, the situation is so new and undeveloped that we cannot get much farther in our analysis than simply asking the right questions; the answers will have to come from our collective experiences and responses as we face these issues in the concrete rather than in the abstract.

Illicit Copying and Use of Software

Trade secret protection, a creation of state common law, has been frequently cited as the best method for protection of the ownership interest in

computer software. Aside from the problems which otherwise limit the effectiveness and scope of this doctrine, it seems totally inapplicable to the use of software in education. That is, educational use is, virtually by definition, the sort of public use or distribution which renders trade secret protection impotent. As a result, most educators encounter the problem of illicit copying and use of software in the context of the federal copyright laws. The present law, the Copyright Act of 1976 (as amended), is encoded as 17 USCA. Under the copyright law, a computer program (or any other work, for that matter) can be copyrighted as soon as it is in a "fixed" form. This means that a program can be copyrighted when it is written or typed on paper, stored on disk or tape, or otherwise stored in a manner in which the program may be read by a person or a machine. The authors' expectations that the program will later be revised or expanded does not affect the copyrightability of the program. Rather, the authors can subsequently copyright the changes and additions as they are made. Thus, if as an educator you are writing a computer program, you will probably want to obtain copyright protection for the program as soon as you create it. Note that the length of protection for programs which are jointly created is the life of the longest-surviving author plus fifty years. However, where you are an educator or a student who wishes to use a program, your interest is in wide use of a work rather than its protection. The copyright laws provide a relatively mechanical solution to this problem; the competing values of encouraging new works and fostering widespread use have been resolved without requiring educators to make ethical choices. The reality, though, is that the ethical values of these solutions have to be communicated to students; respect for federal law may not be enough, particularly when the institution or some of its employees violate the copyright law as a matter of routine.

Legally speaking, a copyright is just that: the exclusive right to make copies of a computer program or other work. The copyright act provides that the owner of a copyright has the exclusive rights to do and to authorize any of the following things:

- To reproduce the program in copies;
- To prepare derivative works based upon the program;
- To distribute copies of the program to the public by sale or other transfer of ownership, or by rental, lease, or lending.

There are also some other rights which mainly involve non-computer works, such as plays and movies. That these rights may also have a basis in the extra-legal societal norms upon which the law is based, then, constitutes the ethical aspect of illicit copying and use.

Unauthorized "System Hacking"

Similarly, there may be few laws regarding the consequences of unauthorized access to a computer system, but we have, I suggest, a common fund of understanding that this is something wrong. The hacker may get some educational value out of this activity, but the wrong created thereby is usually deemed greater than the good. Should this decision be left to the individual, or should a generally accepted sense of the bounds of such activity be organized?

Collection, Use and Maintenance of Information

The creation of data bases large and small, in schools, business, and government, have given rise to articulated fears that misuse of these data bases might infringe on privacy interests. Invasion of privacy is, in this country, overwhelmingly a matter of state common law. The rules thus vary enormously by state; indeed, the right itself was not even recognized until the turn of the century. In the case of data bases, the common law has yet to catch up to our technology. Should the moral values of (non-legal) privacy outweigh the utility of the use of information, in the absence of a rule of law?

A recent book on the growing field of computer law provides information on some of the formal responses which society has made to these problems: Mandell, Steven L. *Computers, Data Processing and the Law*, West Publishing Co. (St. Paul, Minnesota: 1984).